Subject

HAC (Hierarchical agglomerative clustering) – Hybrid Clustering

HAC is a clustering method that produces "natural " groups of examples characterized by attributes. A tree, called dendogram, where successive agglomerations are showed, starting from one example per cluster, until the whole dataset belong to one cluster, describes the clustering process.

The main advantage of HAC is the user can guess the right partitioning by visualizing the tree, he usually prune the tree between nodes presenting an important variation. The main disadvantage is that requires the computation of distances between each example, which is very time consuming when the dataset size increases.

TANAGRA implements a variation of HAC called HYBRID CLUSTERING. Knowing that we need often a very few number of clusters, the construction of the low part of the tree is reserved for a fast method.

There are two steps in the new algorithm:

- First, a low-level clusters are built from fast clustering method such as K-MEANS, SOM;
- HAC starts form these clusters and builds the dendogram.

Note that any clustering algorithm can provide the low level clusters, users can also specify them.

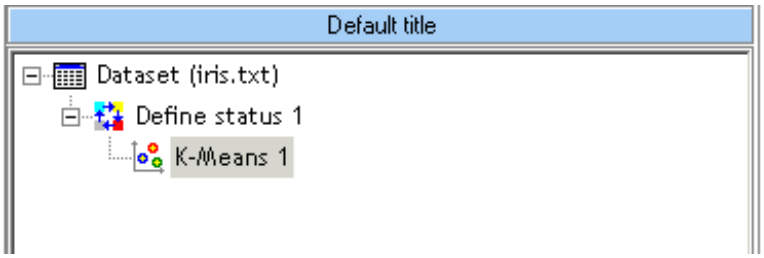
Last, rather than the tree itself, it is the gap between the nodes which is important, these values are provided in a table.

Dataset

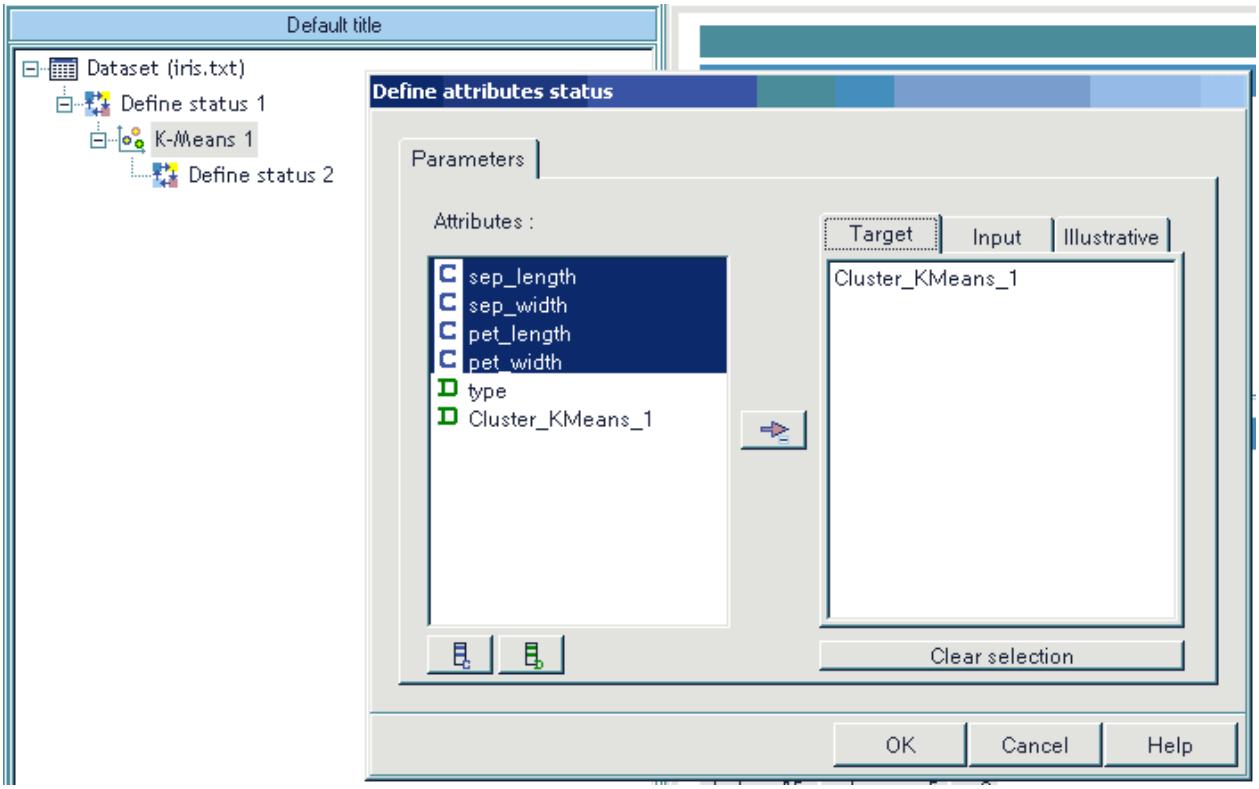
The famous Fisher's IRIS (1936), we know what we must obtain.

HAC process

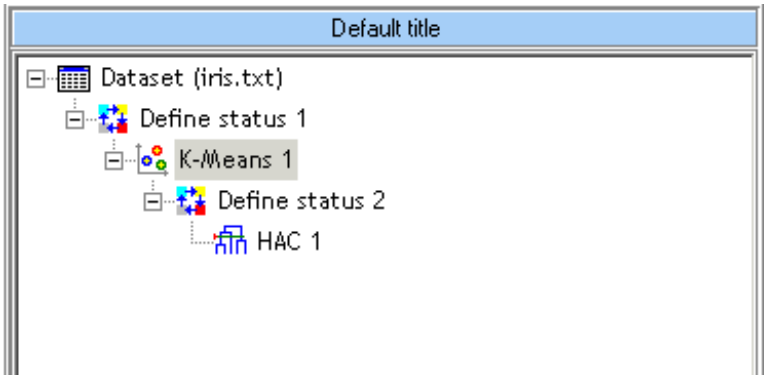
1. Download IRIS_HAC.BDM
2. Insert the "Define Status" component, set all continuous attributes to INPUT
3. Insert into the diagram a K-MEANS component, set the number of classes to 20, leave the other parameters with their default values. ***Run the data mining diagram***, the diagram is the following:



4. Insert a new "Define Status" component, set continuous attributes to INPUT and set Cluster_KMEANS_1, provided by K-MEANS component, as TARGET.



5. Insert HAC component and run the diagram.



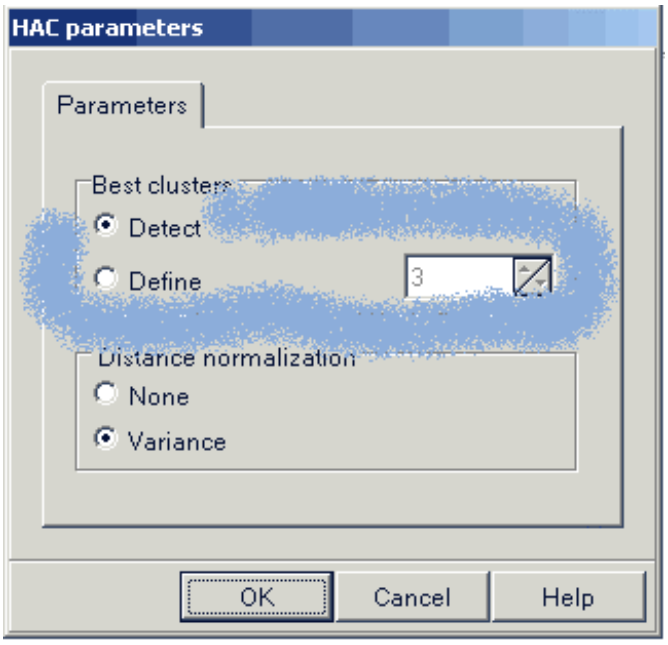
6. Results show tree structure, especially the gap between nodes. The best partitioning, associated with the highest gap, is highlighted.

Best cluster selection

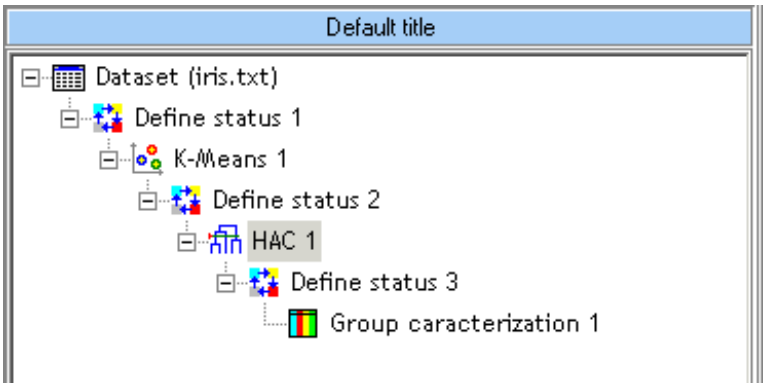
Clusters	BSS ratio	Gap
2	0.6271	2.0102
3	0.7517	0.3247
4	0.7951	0.0026
5	0.8378	0.0897
6	0.8581	0.0146
7	0.8748	0.0011
8	0.8911	0.0060
9	0.9060	0.0045
10	0.9198	0.0272
11	0.9268	0.0067
12	0.9321	0.0023
13	0.9368	0.0019
14	0.9411	0.0045
15	0.9442	0.0013
16	0.9470	0.0001
17	0.9498	0.0006
18	0.9524	0.0040
19	0.9541	0.0022

7. Partitioning into three clusters seems to be powerful in this dataset. There are two comments on theses results:
   - Most of the time, partitioning into two clusters shows always the best gap, it is not interesting. For this reason, it is always ignored but the user can specify it explicitly;
   - In this dataset, the partitioning into three clusters is very significant. But on other datasets, several solutions can be in competition.

8. So, the user can settle himself the right number of clusters



9. Last, clusters must be characterized. On this dataset, which is a particular case, we have the "real" classes membership, it is possible to use them to interpret clusters. Insert a "Define Status" component in the data mining diagram, set "Cluster_HAC_1" as TARGET and "Type" as INPUT. Add a "Group characterization" (Descriptive stats) to the diagram.



10. The results show than clusters correspond to type of IRIS